

# Inheritance

## Employee Management Systems

Developer

Accountant

Supplier

...

# Inheritance

## Employee Management Systems

Developer	
- name	:String
- emailAddress	:String
- phone	:String
- department	:String
- address	:String
- yearOfBirth	:int
- salary	:float
- projectName	:String
..	
..	

Accountant	
- name	:String
- emailAddress	:String
- phone	:String
- department	:String
- address	:String
- yearOfBirth	:int
- salary	:float
- dailyInventory	:String
..	
..	

Supplier	
- name	:String
- emailAddress	:String
- phone	:String
- department	:String
- address	:String
- yearOfBirth	:int
- salary	:float
- city	:String
..	
..	

■ ■ ■ ■

# Inheritance

## Employee Management Systems

Developer	Accountant	Supplier	...
- name :String	- name :String	- name :String	
- emailAddress :String	- emailAddress :String	- emailAddress :String	
- phone :String	- phone :String	- phone :String	
- department :String	- department :String	- department :String	
- address :String	- address :String	- address :String	
- yearOfBirth :int	- yearOfBirth :int	- yearOfBirth :int	
- salary :float	- salary :float	- salary :float	
- projectName :String	- dailyInventory :String	- city :String	
..	..	..	
..	..	..	

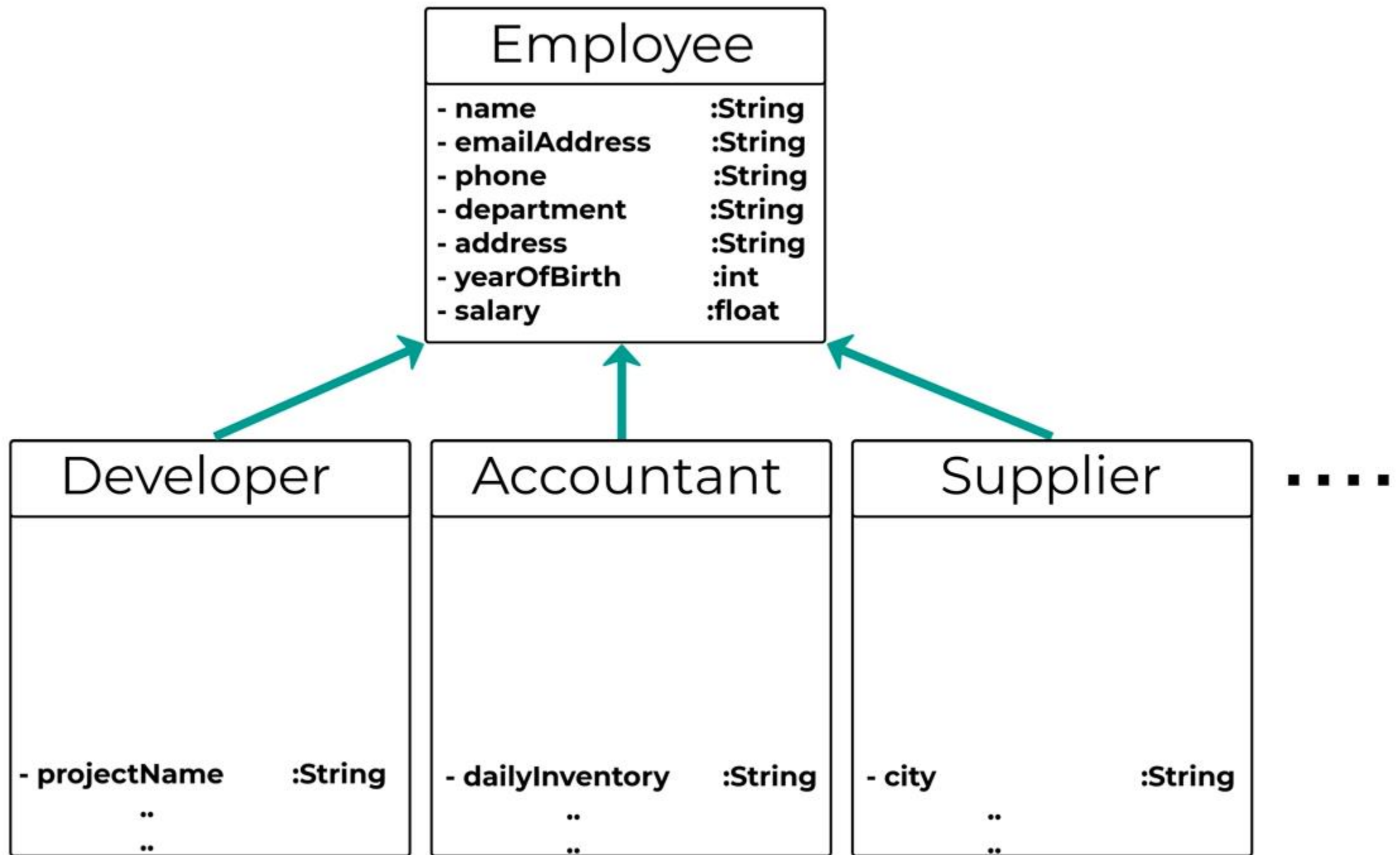
Employee	
- name	:String
- emailAddress	:String
- phone	:String
- department	:String
- address	:String
- yearOfBirth	:int
- salary	:float

Developer	
- projectName	:String
..	
..	

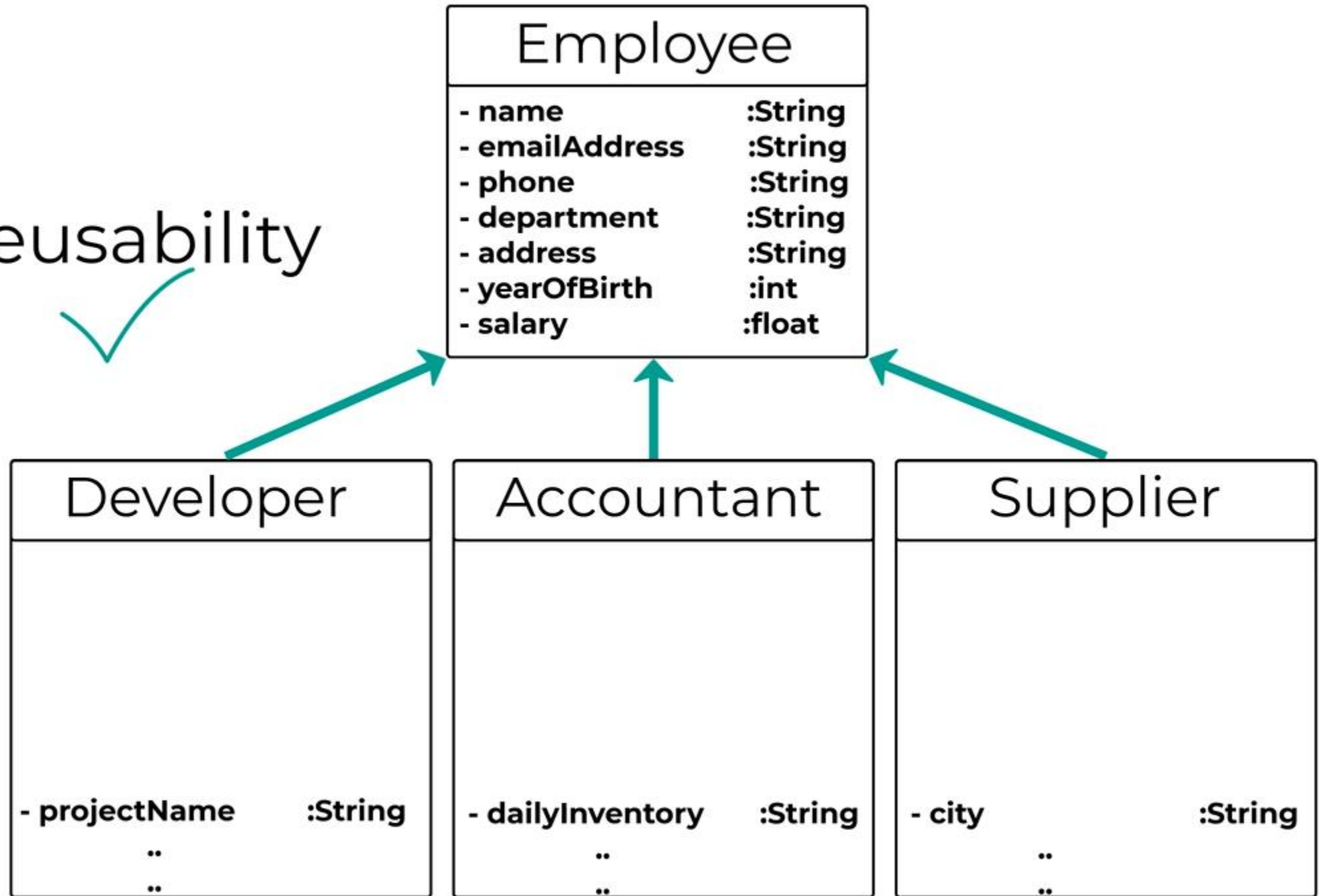
Accountant	
- dailyInventory	:String
..	
..	

Supplier	
- city	:String
..	
..	

■ ■ ■ ■

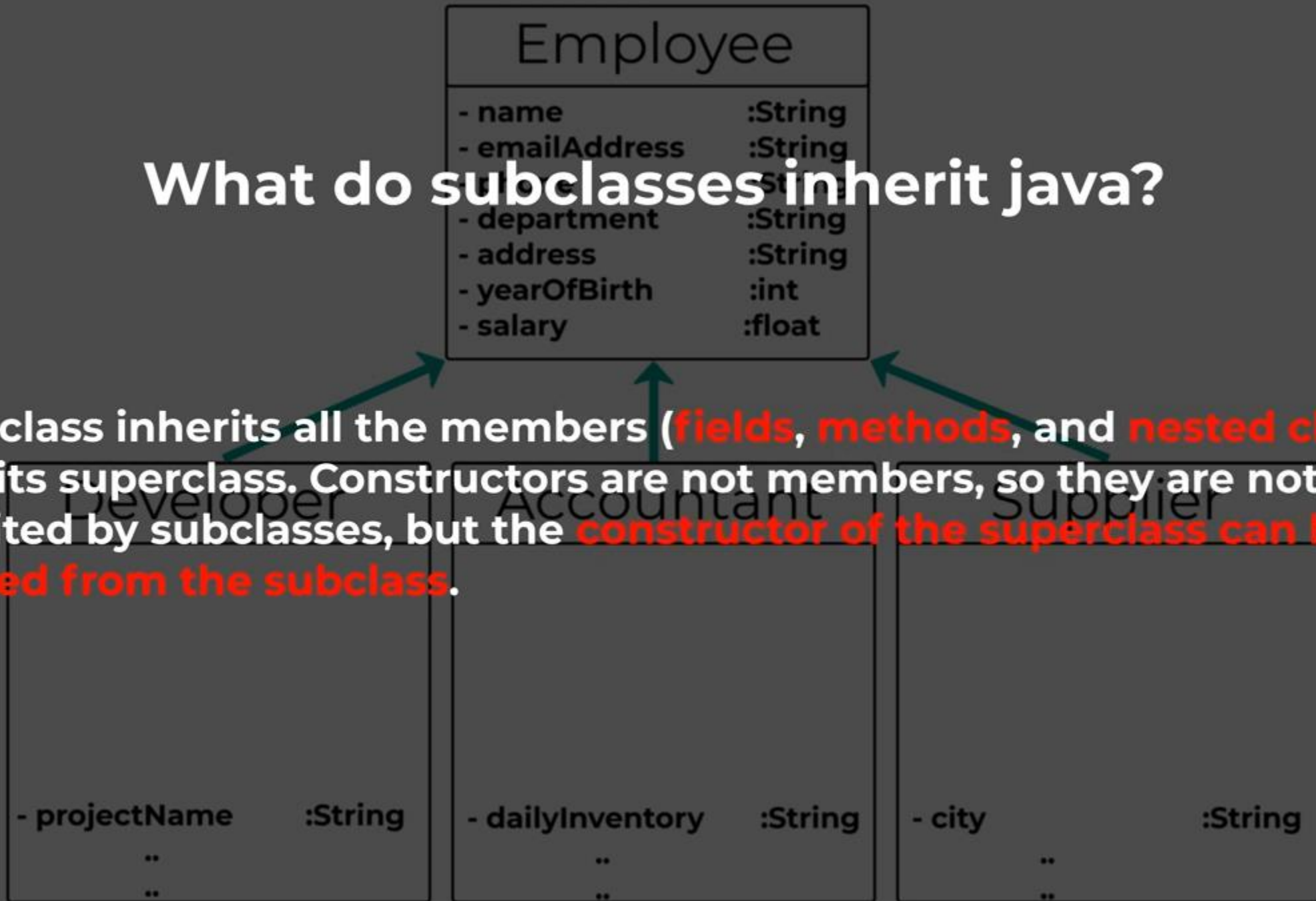


Reusability



# What do subclasses inherit java?

A subclass inherits all the members (**fields, methods, and nested classes**) from its superclass. Constructors are not members, so they are not inherited by subclasses, but the **constructor of the superclass can be invoked from the subclass.**





**Superclass**

**Parent class**

**Base class**

**General class**

**Subclass**

**Child class**

**Derived class**

**Extended class**

Employee

- name :String  
- emailAddress :String  
- phone :String  
- department :String  
- address :String  
- yearOfBirth :int  
- salary :float

Developer

- projectName :String  
..  
..

Accountant

- dailyInventory :String  
..  
..

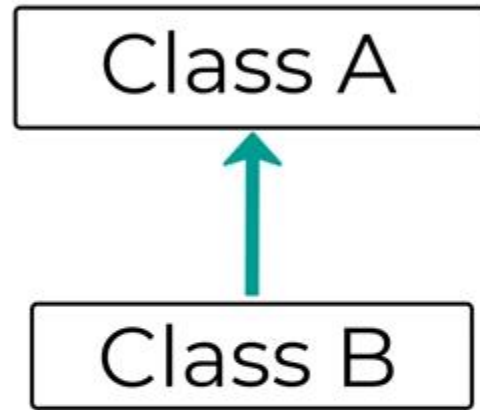
Supplier

- city :String  
..  
..

...

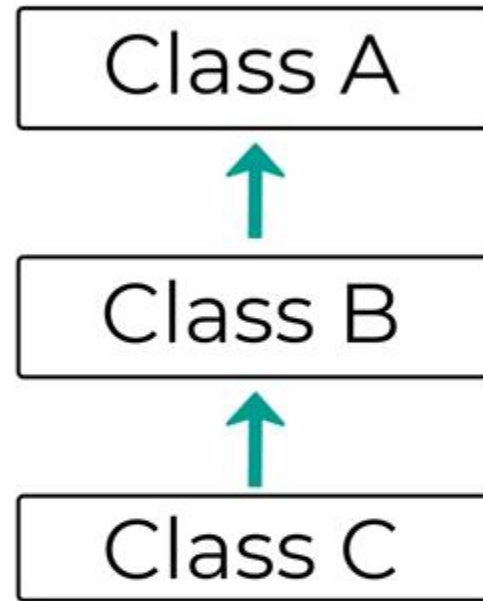


# Types of Inheritance



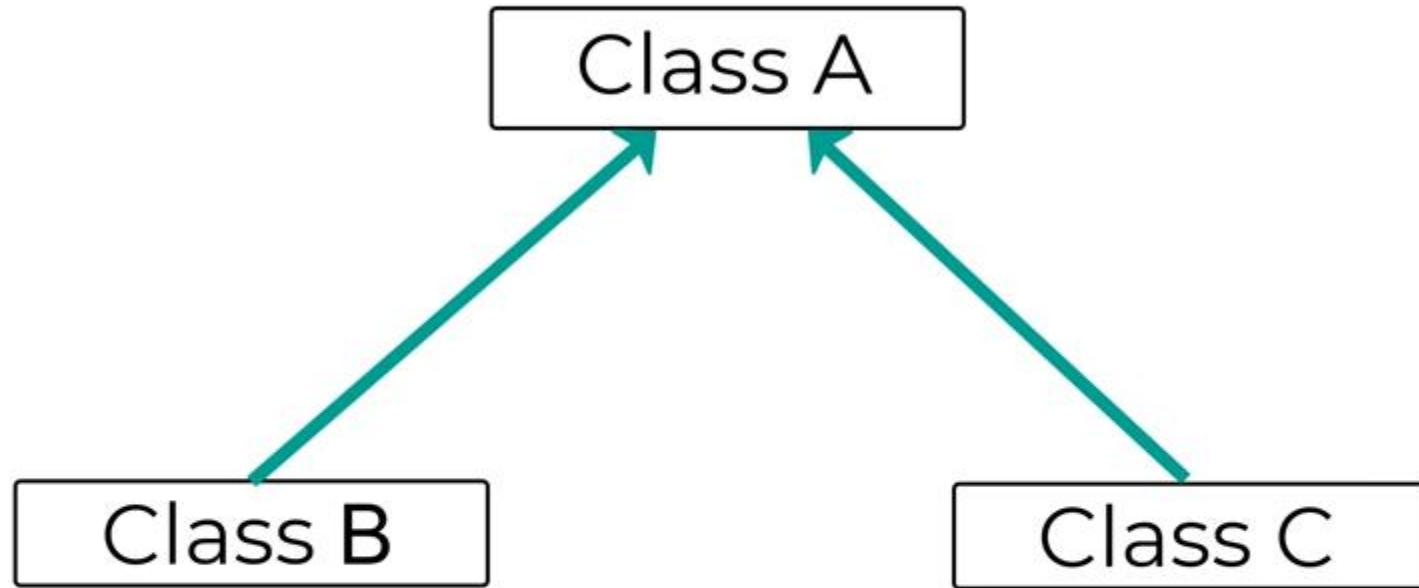
Single Inheritance

# Types of Inheritance



Multilevel Inheritance

# Types of Inheritance



Hierarchical Inheritance

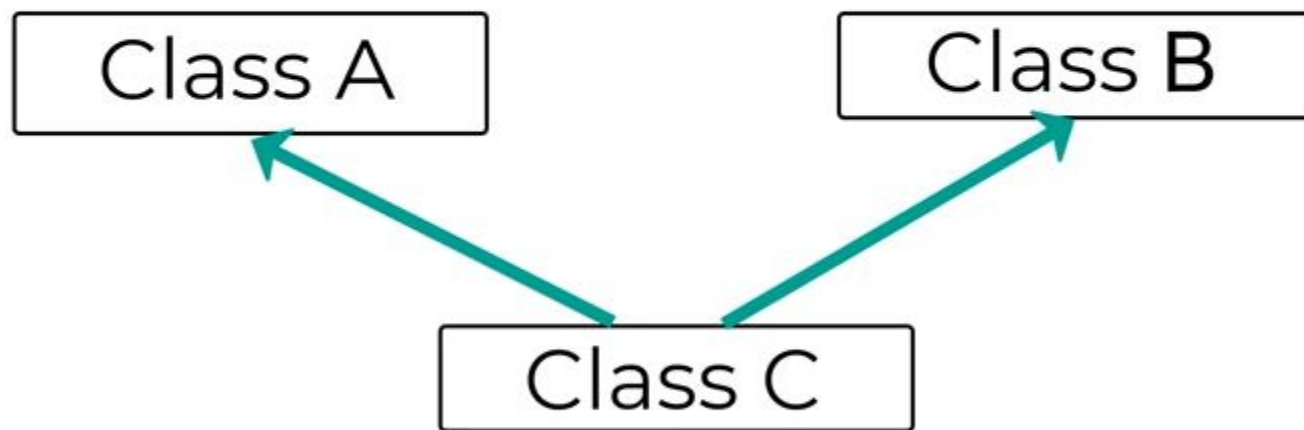
# Types of Inheritance

**Does Java support Multiple inheritance?**

**Multiple inheritances are available in object-oriented programming with C++**

# Types of Inheritance

**Does Java support Multiple inheritance?**



Multiple Inheritance

# Types of Inheritance

**Why multiple inheritance is not supported in java?**

**To reduce the complexity and simplify the language, multiple inheritance is not supported in java.**

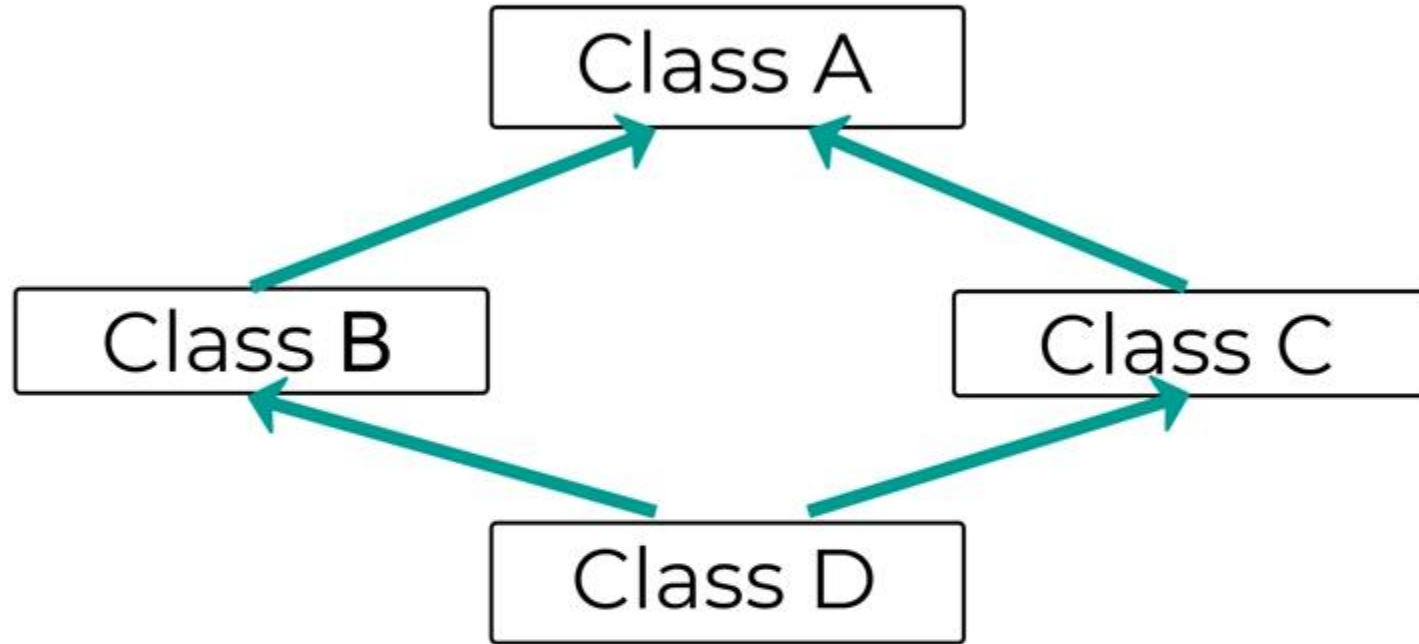
# Types of Inheritance

**Why multiple inheritance is not supported in java?**

Java developers want to use multiple inheritances in some cases. Fortunately, Java developers have **interface** concepts expecting the developers to achieve multiple inheritances by using **multiple interfaces**.



# Types of Inheritance

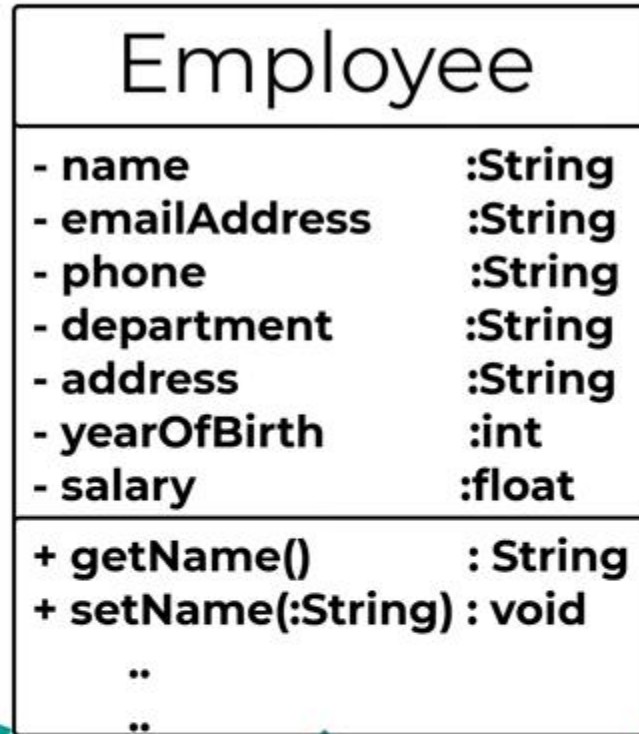


Hybrid Inheritance



# Access Modifiers and Inheritance

**Private:** The access level of a private modifier is **only within the class**. It cannot be accessed from outside the class.



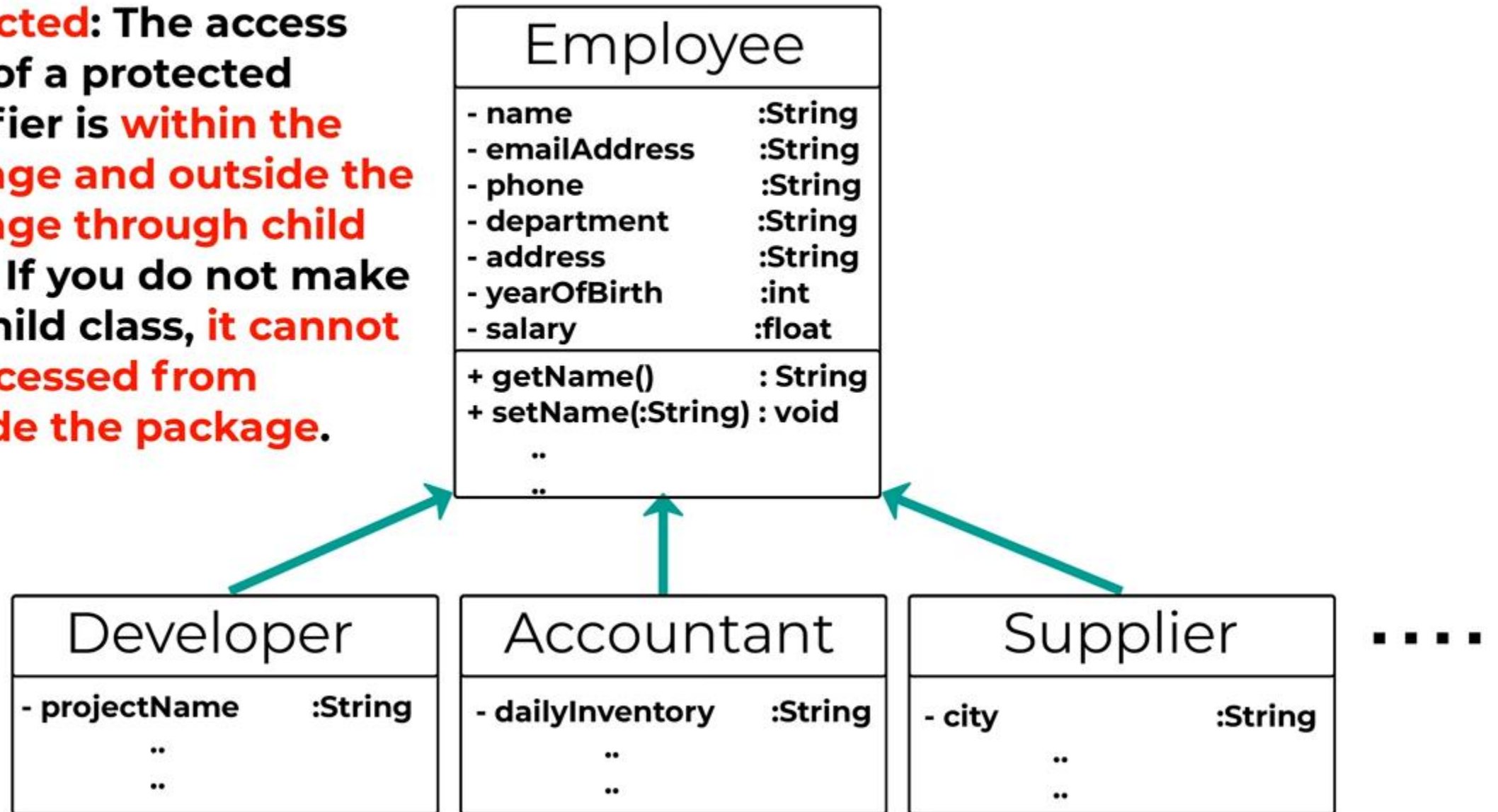
**Public:** The access level of a public modifier is **everywhere**. It can be accessed from within the class, outside the class, within the package and outside the package.



...

# Access Modifiers and Inheritance

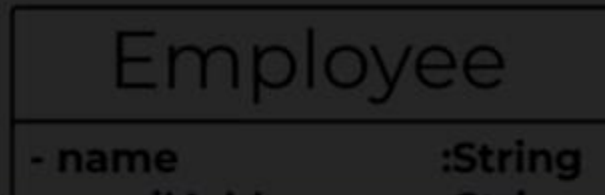
**Protected:** The access level of a protected modifier is **within the package and outside the package through child class**. If you do not make the child class, **it cannot be accessed from outside the package**.



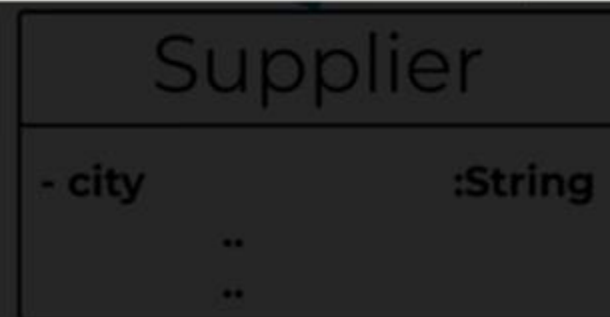
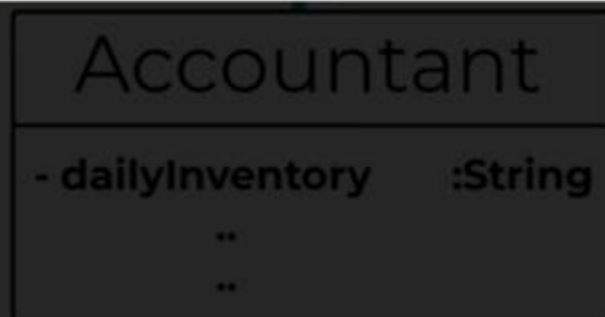


# Access Modifiers and Inheritance

**Protected:** The access level of a protected modifier is **within the**



Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default(package-private)	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y



....

# Employee

**#name :String**  
**#emailAddress :String**  
**#phone :String**  
**#department :String**  
**#address :String**  
**#yearOfBirth :int**  
**#salary :float**

protected

**+ getName() : String**  
**+ setName(:String) : void**  
  
..  
..

Developer

Accountant

Supplier